

前端开发手册

基础概念

- 后端：后端主要负责实现Web App的主要逻辑。例如如果想要实现登陆，那么必然需要后端进行各类设计。为了能够让前端使用具体的功能，后端需要提供api。这种api通常是http协议中的某些方法，通常是GET和POST。例如，为了登陆，前端需要对特定的url调用**POST**方法，同时传入**参数**。
- 前端：前端的代码由浏览器负责执行。比较原始的情况下，浏览器会对html进行渲染，同时执行javascript的代码，有些时候还会用css来渲染某些html元素。接下来默认大家大致接触过以上几种语言。
- vue3框架。vue框架是一个前端框架，他可以使前端的结构变得更加易于管理。其最重要的概念是**组件**，即一个功能完整、显示完全的一个网页部分。通过开发组件而非直接去写html有利于后续工程的推进。对其概念的解释将在后续讲解。

baseline简介

baseline实现了登陆注册功能。由于沟通上的问题，我们的baseline的**UI仍存在问题**，并且UI无助于大家的测试报告，因此暂时来讲大家不需要考虑UI风格的问题。

为了获取baseline的代码，需要从github clone下来。之后大家还需要用git合并分支。

为了运行baseline的代码，需要运行cd到对应目录下执行 `npm run dev`。如果你之前没有装过nodejs的话请自行谷歌对应系统的安装方式。需要注意的是，命令行下应该需要安装nodejs和npm。

之后再本地的浏览器打开命令行中提示的网址即可进入前端页面。大家可以自行玩耍（）

前端代码编写流程

1. 根据需求分析报告和后端api，了解自己的那个任务大概是干嘛的，要做什么。
2. 编写页面。页面的代码应该放在views文件夹下。可以模仿其中的代码。例如：

```
<script setup lang="ts">
import Login from '../components/TheLogin.vue'
</script>

<template>
  <main>
    <div>
      <Login />
    </div>
  </main>
</template>
```

这个文件主要做了1.导入组件 2.使用组件。其余的部分主要是页面渲染代码，暂时不需要注意。在这段代码中，红色下划线的部分就是“组件的使用”。大家可以自行查阅资料理解，这里可以简单理解为红色下划

线部分被直接替换成import进来的东西。

需要注意的是，一个页面可以有多个组件。这里作出要求：**view文件里不允许出现更具体的代码，所有代码必须组合成组件再使用**

3. 将页面注册至router。router文件夹下有一个index文件夹，里面导入了对应的view文件，之后可以照葫芦画瓢把你的页面加入到路由里。这时保存后应该就可以再在浏览器中查看了。例如你注册了“/info”路由，那么在浏览器里输入“http://localhost:5173/info”就可以查看你写的页面，或者看到报错信息。

4. 实现组件。这部分代码要放在components文件夹下。组件包括以下几个部分：

1. 导入需要的组件等。组件里也可以使用组件。之所以说“加入等”是因为有些前端包也需要被导入。例如绝大多数情况下需要导入axios包和vue包，建议这部分直接复制粘贴过来，包括下面axios一行。

```
<script setup lang="ts">
import label_input from './LabelInput.vue';
import { reactive, ref } from 'vue';
import { md5 } from 'js-md5';
import axios from 'axios';
axios.defaults.baseURL = "/api"
```

2. 加入处理函数和存储的数据。大家可以查资料&问gpt&模仿baseline完成处理函数。处理函数需要放到<script>元素里，调用api的话需要使用axios，请参考教程或代码，例如TheLogin文件里的login函数。除了调用api外，代码里也需要定义一些类型（我们使用的是typescript）用于保存用户输入或者接受api返回结果。特别的，有一些数据需要存到浏览器的storage里，如登陆成功的访问令牌，哪些数据需要跨链接使用，就需要存到storage里。

3. 实现模板。在这之后你需要在<template>元素中插入html元素或者vue3代码。这部分是实际会显示的部分，在接受用户输入后可以调用绑定的处理函数。具体的代码要大家自己摸索。

5. 测试功能。

本文所有的描述都不是严格的，因此如果有不理解的概念，需要查询vue3官方中文文档，或者自行百度。